

# An Incremental Knowledge Compilation in First Order Logic

Manoj K. Raut  
 School of Computing Sciences  
 VIT University, Vellore  
 Email: manoj.raut@gmail.com

## Abstract

An algorithm to compute the set of prime implicants of a quantifier-free clausal formula  $X$  in first order logic had been presented in earlier work. As the knowledge base  $X$  is dynamic, new clauses are added to the old knowledge base. In this paper an incremental algorithm is presented to compute the prime implicants of  $X$  and a clause  $C$  from  $\pi(X) \cup C$ . The correctness of the algorithm is also proved.

*Keywords:* Knowledge Compilation, Prime Implicants

## 1 Introduction

Propositional entailment is a central issue in artificial intelligence due to its high complexity. Determining the logical entailment of a given query from a knowledge base is intractable in general [1] as all known algorithms run in time exponential in the size of the given knowledge base. To overcome such computational intractability, the propositional entailment problem is split into two phases such as *off-line* and *on-line*. In the off-line phase the original knowledge base  $X$  is transformed into another knowledge base  $X'$  and the queries are answered in the on-line phase from the new knowledge base in polynomial time in the size of  $X'$ . In such type of compilation most of the computational overhead shifted into the off-line phase is amortized over on-line query answering. The off-line computation is known as *knowledge compilation*.

Several algorithms in knowledge compilation have been suggested so far, see for example, [2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 14, 15]. In these approaches of knowledge compilation, a knowledge base  $KB$  is compiled off-line into

another *equivalent* knowledge base  $\pi(KB)$ , i.e, the set of prime implicates of  $KB$ , so that queries can be answered from  $\pi(KB)$  in polynomial time. Most of the work in knowledge compilation have been restricted to propositional knowledge bases in spite of the greater expressing capacity of a first order knowledge base. Due to lack of expressive power in propositional logic, first order logic is required to represent knowledge in many problems. An algorithm to compute the set of prime implicates of a first order formula in SCNF had been proposed in [10].

As a knowledge base is not static, new clauses are added to the existing knowledge base. It will be inefficient to compute the set of prime implicates of the new knowledge base from the scratch. On the other hand properties of the old  $\pi(KB)$  can be utilized for computing the new  $\pi(KB)$ . In this paper, we suggest an incremental method to compute the set of prime implicates of the new knowledge base from the prime implicates of the old knowledge base. The incremental method based upon the algorithm discussed in [10].

The paper is organized as follows. We present the definitions in Section 2. In Section 3, we describe the properties and main results for computing the prime implicates incrementally. In Section 4, we present the incremental algorithm and its correctness. Section 5 concludes the paper.

## 2 Preliminary Concepts

The alphabet of first order language contains the symbols  $x, y, z, \dots$  as variables,  $f, g, h, \dots$  as function symbols,  $P, Q, R, \dots$  as predicates,  $\neg, \vee, \wedge$  as connectives,  $(, )$  and  $'$  as punctuation marks and,  $\forall$  as the universal quantifier. We assume the syntax and semantics of first order logic. For an interpretation or a first order structure  $i$  and a formula  $X$ , we write  $i \models X$  if  $i$  is a model of  $X$ . For a set of formulas  $\Sigma$  (or a formula) and any formula  $Y$  we write  $\Sigma \models Y$  to denote that for every interpretation  $i$  if  $i$  is a model of every formula in  $\Sigma$  then  $i$  is a model of  $Y$ . In this case, we call  $Y$  a logical consequence of  $\Sigma$ . When  $\Sigma = \{X\}$ , we write  $X \models Y$  instead of  $\{X\} \models Y$ . If  $X \models Y$  and  $Y \models X$  then  $X$  and  $Y$  are said to be equivalent which is denoted by  $X \equiv Y$ .

A literal is an atomic formula or negation of an atomic formula. A disjunctive clause is a finite disjunction of literals which is also represented as a set of literals. A quantifier-free formula is in conjunctive normal form (CNF, infact, SCNF) if it is a finite conjunction of disjunctive clauses. For convenience, a formula is also represented as a set of clauses. In this paper, we consider formulas only in clausal form. In this representation, all variables

are considered universally quantified.

Two literals  $r$  and  $s$  are said to be *complementary* to each other iff the set  $\{r, \neg s\}$  is unifiable with respect to a most general unifier  $\xi$ . We call  $\xi$  a complementary substitution of the set  $\{r, \neg s\}$ . For example,  $Pxf(a)$  and  $\neg Pby$  are complementary to each other with respect to the complementary substitution (most general unifier or mgu, for short)  $[x/b, y/f(a)]$ . So a most general unifier bundles upon infinite number of substitutions to a finite number.

A clause which does not contain a literal and its negation is said to be *fundamental*. Thus a non-fundamental clause is valid. We avoid taking non-fundamental clauses in clausal form because the universal quantifiers appearing in the beginning of the formula can appear before each conjunct of the CNF since  $\forall$  distributes over  $\wedge$ . So each clause in a formula of the knowledge base is assumed to be non-fundamental.

Let  $C_1$  and  $C_2$  be two disjunctive clauses. Then  $C_1$  *subsumes*  $C_2$  iff there is a substitution  $\sigma$  such that  $C_1\sigma \subseteq C_2$ , i.e.,  $C_1\sigma \models C_2$ . For example,  $\{\neg Rxf(a), \neg Py\}$  subsumes  $\{\neg Rg(a)f(a), \neg Py, Qz\}$  with respect to a  $\sigma = [x/g(a)]$ . A disjunctive clause  $C$  is an *implicate* of a finite set of formulas  $X$  (assumed to be in CNF) if  $X\sigma \models C$  for a substitution  $\sigma$ . We write  $I(X)$  as the set of all implicates of  $X$ . A clause  $C$  is a *prime implicate* of  $X$  if  $C$  is an implicate of  $X$  and there is no other implicate  $C'$  of  $X$  such that  $C'\tau \models C$  for a substitution  $\tau$  (i.e., if no other implicate  $C'$  subsumes  $C$ ).  $\Pi(X)$  denotes the set of all prime implicates of  $X$ . It may be observed that if  $C$  is not prime then there exists a prime implicate  $D$  of  $X$  such that  $D\tau \models C$ . The set of all implicates of  $X$  is denoted by  $\Psi(X)$ .

Note that the notion of prime implicate is well defined as the knowledge base contains clauses unique up to subsumption. Let  $Y$  be a set of fundamental clauses. The residue of subsumption of  $Y$ , denoted by  $Res(Y)$  is a subset of  $Y$  such that for every clause  $C \in Y$ , there is a clause  $D \in Res(Y)$  where  $D$  subsumes  $C$ ; and no clause in  $Res(Y)$  subsumes any other clause in  $Res(Y)$ .

Let  $C_1, C_2$  be two clauses in  $X$  and  $r \in C_1, s \in C_2$  be a pair of complementary literals with respect to a most general unifier  $\sigma$ . The resolution of  $C_1$  and  $C_2$  is  $C = [(C_1 - \{r\}) \cup (C_2 - \{s\})]\sigma$ . If  $C$  is fundamental, it is called *consensus* of  $C_1$  and  $C_2$  denoted by  $CON(C_1, C_2)$ .  $C$  can also be written as  $[(C_1\sigma - \{t\}) \cup (C_2\sigma - \{\neg t\})]$  for a literal  $t$ , provided  $r\sigma = t$  and  $s\sigma = \neg t$ . We can also say that  $C$  is the propositional consensus of  $C_1\sigma$  and  $C_2\sigma$ . For example, if  $C_1 = \{Rbx, \neg Qg(a)\}$  and  $C_2 = \{Rab, Qz\}$  then  $CON(C_1, C_2) = \{Rbx, Rab\} =$  propositional consensus of  $C_1[z/g(a)]$  and  $C_2[z/g(a)]$ . If  $C$  is the consensus of  $C_1$  and  $C_2$  with respect to a most

general unifier  $\sigma$  then  $C$  is said to be associated with  $\sigma$ . By default, each clause  $C$  of a set of formulas  $X$  is associated with the empty substitution  $\epsilon$ . Let  $C_1$  and  $C_2$  be two resolvent clauses associated with substitutions  $\sigma_1$  and  $\sigma_2$ , respectively. Then their consensus with respect to  $\sigma$  is defined provided  $\sigma_1\sigma = \sigma_2\sigma$ . In that case the consensus is the propositional consensus of  $C_1\sigma$  and  $C_2\sigma$  and the consensus is associated with the substitution  $\sigma_3 = \sigma_1\sigma = \sigma_2\sigma$ .

### 3 Computation of Prime Implicates

Besides presenting some main results from [10], we describe the computation of prime implicates incrementally of quantifier free first order formulas in clausal form. Let  $X = \{C_1, \dots, C_n\}$  be a formula where each disjunctive clause  $C_i$  is fundamental. Each  $C_i$  is an implicate of  $X$  with respect to the empty substitution, but each may not be a prime implicate. The key is the subsumption of implicates of  $X$ . As the clauses we deal with are disjunctive, if  $C_1$  subsumes  $C_2$  then there is a substitution  $\sigma$  such that  $C_1\sigma \models C_2$ . We will see that computation of prime implicates is the result of deletion of subsumed clauses from the consensus closure. We also explore the relationship between consensus closure and prime implicates of a formula. We can derive the following two results from [10].

**Lemma 3.1** *A clause  $D$  is an implicate of  $X \cup C$  if and only if there is a prime implicate  $D'$  of  $X \cup C$  such that  $D'$  subsumes  $D$ .*

**Lemma 3.2**  $X \cup C \equiv \Psi(X \cup C) \equiv \Pi(X \cup C)$

The computational aspects of prime implicates is given below. For a set of clauses  $X$ , let  $L(X)$  be the set of all consensus among clauses in  $X$  along with the clauses of  $X$ , i.e.,  $L(X) = X \cup \{S : S \text{ is a consensus of each possible pair of clauses in } X\}$ . We construct the sequence  $X, L(X), L(L(X)), \dots$ , i.e.,  $L^0(X) = X$ ,  $L^{n+1}(X) = L(L^n(X))$  for  $n \geq 0$ . Define the *consensus closure* of  $X$  as  $\bar{L}(X) = \cup\{L^i(X) : i \in \mathbb{N}\}$ .

**Example 3.1** Let  $X = (Pxa \vee \neg Qaf(x)) \wedge (\neg Pba \vee Rbz) \wedge (\neg Rxf(a) \vee Qzf(a)) = C_1 \wedge C_2 \wedge C_3$ .

The consensus of  $C_1$  and  $C_2$  with respect to the substitution  $[x/b]$  is  $C_4 = (\neg Qaf(b) \vee Rbz)$ , of  $C_1$  and  $C_3$  with respect to the substitution  $[z/a, x/a]$  is  $C_5 = (Paa \vee \neg Raf(a))$ , of  $C_2$  and  $C_3$  with respect to the

substitution  $[x/b, z/f(a)]$  is  $C_6 = (\neg Pba \vee Qf(a)f(a))$ . So  $C_4, C_5, C_6$  are associated with substitutions  $[x/b], [z/a, x/a], [x/b, z/f(a)]$  and each of  $C_1, C_2, C_3$  is associated with the empty substitution  $\epsilon$ . Then

$$\begin{aligned} L^1(X) &= (Pxa \vee \neg Qaf(x)) \wedge (\neg Pba \vee Rbz) \wedge (\neg Rxf(a) \vee Qzf(a)) \wedge \\ &\quad (\neg Qaf(b) \vee Rbz) \wedge (Paa \vee \neg Raf(a)) \wedge (\neg Pba \vee Qf(a)f(a)) \\ &= C_1 \wedge C_2 \wedge C_3 \wedge C_4 \wedge C_5 \wedge C_6. \end{aligned}$$

The consensus of  $C_3$  and  $C_4$  with respect to the substitution  $[x/b, z/f(a)]$  is  $C_7 = (Qf(a)f(a) \vee \neg Qaf(b))$ . So  $C_7$  is associated with the substitution  $[x/b, z/f(a)]$ . Note that the consensus of  $C_1$  and  $C_6$  with respect to the substitution  $[x/b]$  is not possible as the composition of substitution is not well defined, i.e.,  $[\epsilon][x/b] \neq [x/b, z/f(a)][x/b]$ . Similarly the consensus between  $C_3$  and  $C_5$  is not possible as the composition of substitution is not well defined. Hence,

$$\begin{aligned} L^2(X) &= ((Pxa \vee \neg Qaf(x)) \wedge (\neg Pba \vee Rbz) \wedge (\neg Rxf(a) \vee Qzf(a)) \wedge (\neg Qaf(b) \vee \\ &\quad Rbz) \wedge (Paa \vee \neg Raf(a)) \wedge (\neg Pba \vee Qf(a)f(a)) \wedge (Qf(a)f(a) \vee \neg Qaf(b))). \end{aligned}$$

Since  $L^2(X) = L^3(X)$ , we have,  $\overline{L}(X) = L^2(X)$ .  $\square$

As each clause of a formula  $X$  is itself an implicate, the following result shows that other implicates can be computed by taking consensus among the clauses of a formula  $X$ .

**Theorem 3.1** *Consensus of two implicates of  $\pi(X_1) \cup X_2$  is an implicate of the formula  $X_1 \cup X_2$  where  $X_1$  and  $X_2$  are sets of clauses.*

**Proof.** let  $C_1$  and  $C_2$  be two implicates of  $\pi(X_1) \cup X_2$  associated with  $\sigma_1$  and  $\sigma_2$  respectively.  $(\pi(X_1) \cup X_2)\sigma_1 \models C_1$ ,  $(\pi(X_1) \cup X_2)\sigma_2 \models C_2$ .  $CON(C_1, C_2) = PCON(C_1\sigma, C_2\sigma)$  provided  $\sigma_1\sigma = \sigma_2\sigma$  for some substitution  $\sigma$ . So  $C = CON(C_1, C_2) = ((C_1\sigma - \{t\}) \cup (C_2\sigma - \{\neg t\}))$ . Let  $i \models (X_1 \cup X_2)$ .  $i \models (X_1 \cup X_2)\sigma = X_1\sigma \cup X_2\sigma$ . So  $i \models X_1\sigma$  or  $i \models X_2\sigma$ . Let  $i \models X_1\sigma$ . Then,  $i \models \pi(X_1)\sigma$  (by Lemma 3.2. Moreover,  $i \models \pi(X_1)\sigma \cup X_2\sigma$ .  $i \models (\pi(X_1) \cup X_2)\sigma$ .  $i \models C_1$  and  $i \models C_1\sigma$ . Similarly, if  $i \models X_2\sigma$  then  $i \models C_2\sigma$ . Suppose  $i \models t$ . Then  $i \models C_2\sigma - \{\neg t\}$ , i.e.,  $i \models C$ . Similarly, if  $i \models \neg t$ , then  $i \models C_1\sigma - \{t\}$ , i.e.,  $i \models C$ . Hence  $C$  is an implicate of  $X_1 \cup X_2$ .  $\square$

The following result tells that we can add the prime implicates one by one to  $X$  as it preserves the truth value.

**Theorem 3.2** *Let  $X = \{C_1, C_2, \dots, C_n\}$  be a formula and  $C$  be the consensus of a pair of clauses from  $X$  then  $X \equiv X \wedge C$ .*

**Proof.** As  $X$  is in SCNF and  $C$  is a disjunctive clause,  $X \wedge C \models X$ . Conversely, let  $i \models X$ . Then  $i \models C_k$  for  $1 \leq k \leq n$ , and  $i \models C_i \wedge C_j$  for  $1 \leq \{i, j\} \leq n$ . Let  $C = CON(C_i, C_j)$ , where  $r$  and  $s$  be a pair of complementary literals and  $r \in C_i$ ,  $s \in C_j$ ,  $r\sigma = t$  and  $s\sigma = \neg t$ . Let  $i \models C_i \wedge C_j = (D_1 \vee r) \wedge (D_2 \vee s)$ , where  $D_1$  is a disjunctive clause in  $C_i$  leaving  $r$  and  $D_2$  is a disjunctive clause from  $C_j$  leaving  $s$ .  $C = CON(C_i, C_j) = CON(D_1 \vee r, D_2 \vee s) = D_1\sigma \vee D_2\sigma$ . Further,  $i \models (D_1 \vee r)$  and  $i \models (D_2 \vee s)$ . If  $i \models D_1 \vee r$  then  $i \models D_1$ .  $i \models D_1\sigma$ .  $i \models D_1\sigma \vee D_2\sigma$  and  $i \models C$ . Similarly if  $i \models D_2 \vee s$  then  $i \models C$ . Let  $i \models r$ . Then  $i \not\models s$ , so  $i \models D_2$  and  $i \models D_2\sigma$ . This implies,  $i \models C$ . If  $i \models s$  then  $i \not\models r$ ,  $i \models D_1$ .  $i \models D_1\sigma$ , and  $i \models C$ . This implies  $i \models X \wedge C$ .  $\square$

**Theorem 3.3**  $Res(\overline{L}(\pi(X) \cup C)) = Res(\overline{L}(X \cup C))$

**Proof.** Obviously,  $\overline{L}(\pi(X) \cup C) \subseteq \overline{L}(X \cup C)$ . This implies  $Res(\overline{L}(\pi(X) \cup C)) \subseteq Res(\overline{L}(X \cup C))$

Conversely, let  $C_1 \in Res(\overline{L}(X \cup C))$ . So  $C_1 \in \overline{L}(X \cup C)$  and there exists no  $D \in \overline{L}(X \cup C)$  such that  $D$  subsumes  $C_1$ . If  $C_1 \notin Res(\overline{L}(\pi(X) \cup C))$  then  $D_1 \in \overline{L}(\pi(X) \cup C)$  such that  $D_1$  subsumes  $C_1$ . As  $\overline{L}(\pi(X) \cup C) \subseteq \overline{L}(X \cup C)$ , there exists  $D_1 \in \overline{L}(X \cup C)$  such that  $D_1$  subsumes  $C_1$ . This gives a contradiction. So  $C_1 \in Res(\overline{L}(\pi(X) \cup C))$ . This implies  $\overline{L}(X \cup C) \subseteq Res(\overline{L}(\pi(X) \cup C))$ .  $\square$

The following results are consequences of Lemma 3.1 and 3.2 and Theorem 3.3.

**Theorem 3.4** *A clause  $D_1$  is an implicate of  $X \cup C$  iff there is  $D_2 \in \overline{L}(\pi(X) \cup C)$  such that  $D_2$  subsumes  $D_1$ .*

**Theorem 3.5** *The set of all prime implicates of  $X \cup C$  is a subset of the consensus closure of  $\pi(X) \cup C$ , i.e.,  $\pi(X \cup C) \subseteq \overline{L}(\pi(X) \cup C)$ . Moreover,  $\pi(X \cup C) = Res(\overline{L}(\pi(X) \cup C))$*

Moreover, the sets  $\pi(X_1 \cup X_2)$  and  $\pi(\pi(X_1) \cup X_2)$  are not only equivalent but also identical, as the next result shows. theorem.

**Theorem 3.6**  $\pi(X_1 \cup X_2) = \pi(\pi(X_1) \cup X_2)$

**Proof.** let  $C \in \pi(X_1 \cup X_2)$ .  $(X_1 \cup X_2)\sigma_1 \models C$  and there does not exist any implicate  $D$  of  $X_1 \cup X_2$  such that  $D$  subsumes  $C$ .  $X_1\sigma \cup X_2\sigma \models C$  and there doesnot exist any implicate  $D$  of  $X_1 \cup X_2$  such that  $D$  subsumes  $C$ . As  $X_1 \equiv \pi(X_1)$ ,  $\pi(X_1)\sigma \cup X_2\sigma \models C$  and there does not exist any implicate  $D$  of  $\pi(X_1) \cup X_2$  such that  $D$  subsumes  $C$ .  $(\pi(X_1) \cup X_2)\sigma \models C$  and there does not exist any implicate  $D$  of  $\pi(X_1) \cup X_2$  such that  $D$  subsumes  $C$ .  $C \in \pi(\pi(X_1) \cup X_2)$ . So  $\pi(X_1 \cup X_2) \subseteq \pi(\pi(X_1) \cup X_2)$ . Similarly the inclusion  $\pi(\pi(X_1) \cup X_2) \subseteq \pi(X_1 \cup X_2)$  is proved.  $\square$

## 4 Incremental Algorithm

The following algorithm computes the set of prime implicates of  $\pi(X_1) \wedge \Sigma$  (i.e, of  $(\pi(X_1) \cup \Sigma)$  by consensus subsumption method in first order logic. Recall that for a set of clauses  $A$ ,  $L(A)$  denotes the set of clauses of  $A$  along with the consensus of each possible pair of clauses of  $A$ . The algorithm computes the consensus  $L(\pi(X_1) \cup \Sigma)$  by taking clauses from  $\pi(X_1)$  and clauses from  $\Sigma$ . It does not compute the consensus between two clauses of  $\pi(X_1)$  as it is wasteful.  $L(\pi(X_1) \wedge \Sigma) = \{CON(D_1, D_2) : D_1 \in \pi(X_1) \cup \Sigma \text{ and } D_2 \in \Sigma\}$ . The algorithm applies subsumption on  $L(\pi(X_1) \wedge \Sigma)$  and keeps the residue  $Res(L(\pi(X_1) \wedge \Sigma))$  and repeat the steps till two iteration steps produce the same result.

**Algorithm** *INCRPI*

Input: The set of prime implicates  $\pi(X_1)$  and a clause  $C$

Output: The set of prime implicates of  $X_1 \cup C$

```

begin
  if  $C$  is a non-fundamental clause, then
     $\pi(X_1 \cup C) = \pi(X_1)$ 
  else
     $\Sigma = \{C\}$ ;
     $\eta_0 = \emptyset$ ;
     $i = 1$ ;
     $\gamma = \pi(X_1) \cup \Sigma$ ;
     $\eta_i = \text{Res}(\pi(X_1) \cup \Sigma)$ ;
    if  $\Sigma$  is deleted
      then stop
    else
      while  $\eta_i \neq \eta_{i-1}$ 
        compute  $R = \text{CON}(D_1, D_2)$  s.t.  $D_1 \in \eta_i$  and  $D_2 \in \Sigma$ ;
         $\Sigma = \Sigma \cup R$ ;
         $L(\eta_i) = \eta_i \cup \Sigma$ ;
         $\eta_{i+1} = \text{Res}(L(\eta_i))$ ;
        if any clause of  $\Sigma$  is deleted
          then update  $\Sigma$ ;
         $i = i + 1$ ;
       $\pi(\gamma) = \eta_{i+1}$ ;
end

```

**Theorem 4.1** *Let  $\pi(X_1)$  be a set of prime implicates of a formula  $X_1$  and  $C$  be any clause. The incremental algorithm generates the set of prime implicates of  $X_1 \cup \{C\}$ .*

**Proof.** Let  $\gamma = \pi(X_1) \cup \{C\}$ .  $\eta_1$  is computed by taking residue of subsumption on  $\gamma$ . If a clause  $C \in \gamma$  subsumes a clause  $D \in \gamma$ , any clause entailed by  $D$  is entailed by  $C$ . So  $D$  can be discarded from  $\gamma$  without changing its deduction closure. Let  $\eta_1 = \text{Res}(\gamma)$ .  $\gamma = \pi(\eta_1)$ , by Theorem 3.5.

If the clause  $C$  is deleted while taking residue then  $\pi(\gamma) = \pi(X_1 \cup \{C\}) = \pi(X_1)$ . If  $C$  is not deleted from  $\Sigma$ , then the algorithm computes the consensus  $R$  between a pair of clauses from  $\eta_1$  (i.e.,  $\text{Res}(\gamma)$ ) and  $C$ . It does not take consensus between two clauses in  $\pi(X_1)$  as they are prime implicates. Any attempt to take consensus between them will increase the number of unnecessary operations. As every clause of  $\eta_1$  is an implicate of  $\eta_1$ , by Theorem



3.1,  $R$  is an implicate of  $\eta_1$ , i.e., of  $\gamma$ .  $R$  can be added to  $\eta_1$  without changing its deduction closure by Theorem 3.2. We add  $R$  to  $\Sigma = \{C\}$  as the new clauses formed can also take part in further consensus. We maintain  $\pi(X_1)$  and  $\Sigma$  separately so that while taking consensus next time at least one clause will be from  $\Sigma$ , i.e.,  $L(\eta_1) = \eta_1 \cup \Sigma$ . By Theorem 3.5,  $\pi(\gamma) = \pi(\eta_1) \subseteq \overline{L}(\eta_1)$ . If any clause  $C$  subsumes a clause  $D$  in  $L(\eta_1)$  then  $D$  can be discarded without changing the deduction closure as  $C \models D\sigma$ . Note  $\eta_2 = \text{Res}(L(\eta_1))$ . By Theorem 3.5,  $\pi(\gamma) = \pi(\eta_1) \subseteq \overline{L}(\eta_2)$ . In general,  $\pi(\gamma) = \pi(\eta_1) \subseteq \eta_i = \text{Res}(L(\eta_{i-1})) \subseteq \overline{L}(\eta_{i-1})$ . If the algorithm terminates, at some stage then  $\overline{L}(\eta_i) = \overline{L}(\eta_{i-1})$  and  $\text{Res}(\overline{L}(\eta_i)) = \text{Res}(\overline{L}(\eta_{i-1}))$ . By Theorem 3.5,  $\pi(\gamma) = \text{Res}(\overline{L}(\eta_i))$ . Since the algorithm computes  $\eta_{i+1} = \text{Res}(L(\eta_i))$ ,  $\pi(\gamma) = \eta_{i+1}$ . By Theorem 3.6 we obtain the set of prime implicates of  $X_1 \cup C$  as  $\pi(X_1 \cup C) = \pi(\pi(X_1) \cup C) = \pi(\gamma) = \eta_{i+1}$ .  $\square$

**Example 4.1** Let  $X = \{\{Qy\}, \{\neg Rf(x)b\}, \{Px \vee Rzb \vee \neg Qz\}\} = \eta_1$  (say) and  $C$  be the clause  $C = \{\neg Pa \vee \neg Qz\} = \Sigma$ , another clause. Take  $\Sigma = \{C\}$

As computed in [10], the set of prime implicates of the formula  $X$  is  $\pi(X) = \{\{Qy\}, \{\neg Rf(x)b\}, \{Px \vee Rzb\}, \{Px \vee \neg Qz\}\}$ , where the clause  $Qy$  is associated with  $\epsilon$ ,  $\neg Rf(x)b$  is associated with  $\epsilon$ ,  $Px \vee Rzb$  is associated with  $[y/z]$  and  $Px \vee \neg Qz$  is associated with  $[y/f(x)]$ .

Let  $\eta_1 = \pi(X) \cup \Sigma = \{\{Qy\}, \{\neg Rf(x)b\}, \{Px \vee Rzb\}, \{Px \vee \neg Qz\}, \{\neg Pa \vee \neg Qz\}\}$ .

As the literal  $Qy$  in  $\{Qy\}$  and  $\neg Qz$  in  $\{\neg Pa \vee \neg Qz\}$  are a pair of complementary literals with respect to the substitution  $[y/z]$ , the consensus between the clauses  $\{Qy\}$  and  $\{\neg Pa \vee \neg Qz\}$  is  $\{\neg Pa\}$ . Here, the substitution  $[y/z]$  is a most general unifier. Note that we can not take consensus between  $\{Px \vee Rzb\}$  and  $\{\neg Pa \vee \neg Qz\}$ , between  $\{Px \vee \neg Qz\}$  and  $\{\neg Pa \vee \neg Qz\}$  as composition of substitution are not well defined. Now updating  $\Sigma$  we get,  $\Sigma = \{\{\neg Pa \vee \neg Qz\}, \{\neg Pa\}\}$ . The new clause formed is added to  $\eta_1$  to get  $L(\eta_1)$ .

$L(\eta_1) = \{\{Qy\}, \{\neg Rf(x)b\}, \{Px \vee Rzb\}, \{Px \vee \neg Qz\}, \{\neg Pa \vee \neg Qz\}, \{\neg Pa\}\}$ .

As  $\{\neg Pa\}$  subsumes  $\{\neg Pa \vee \neg Qz\}$  in  $L(\eta_1)$ , we get the residue as

$\eta_2 = \text{Res}(L(\eta_1)) = \{\{Qy\}, \{\neg Rf(x)b\}, \{Px \vee Rzb\}, \{Px \vee \neg Qz\}, \{\neg Pa\}\}$ .

Now the clause  $\{Px \vee Rzb\}$  associated with  $[y/z]$  and  $\{\neg Pa\}$  associated with  $[y/z]$  contain a pair of complementary literals. The consensus between  $\{Px \vee Rzb\}$  and  $\{\neg Pa\}$  with respect to the substitution (mgu)  $[y/z, x/a]$  is  $Rzb$ . Again we can not take consensus between  $\{Px \vee \neg Qz\}$  associated with  $[y/f(x)]$  and  $\{\neg Pa\}$  associated with  $[y/z]$  as composition of substitution is not well defined. Now  $\Sigma = \{\{\neg Pa\}, \{Rzb\}\}$ . We now add the new clause to  $\eta_2$  to get  $L(\eta_2)$ .

$$L(\eta_2) = \{\{Qy\}, \{\neg Rf(x)b\}, \{Px \vee Rzb\}, \{Px \vee \neg Qz\}, \{\neg Pa, Rzb\}\}.$$

As  $\{Rzb\}$  subsumes  $\{Px \vee Rzb\}$ , the residue becomes

$$\eta_3 = \text{Res}(L(\eta_2)) = \{\{Qy\}, \{\neg Rf(x)b\}, \{Px \vee \neg Qz\}, \{\neg Pa\}, \{Rzb\}\}.$$

We cannot take any more consensus among the clauses of  $\eta_3$  as the composition of substitution is not well defined between clauses. So  $\eta_3 = L(\eta_3) = \eta_4$ . That is,

$$\pi(X \cup C) = \pi(\pi(X) \cup C) = \{\{Qy\}, \{\neg Rf(x)b\}, \{Px \vee \neg Qz\}, \{\neg Pa\}, \{Rzb\}\}.$$

## 5 Conclusion

In this paper, we have suggested an incremental algorithm to compute the set of prime implicates of a knowledge base  $X$  and a clause  $C$ . We have also proved the correctness of the algorithm. In Example 4.1, when new clauses or clause sets are added, computation of the prime implicates uses the primeness of the already computed clauses. The algorithm adds one clause at a time and compiles the enhanced knowledge base. A simple modification of the algorithm can be made to accomodate a set of clauses instead of one by putting *INCRPI* inside an iterative loop.

If we compute the prime implicates of  $X \cup C$  directly by using the algorithm from [10], we obtain the same prime implicates, though it involves wasteful computations. Efficiency of the proposed algorithm *INCRPI* results in exploiting the properties of  $\pi(X)$  instead of using  $X$  directly.

## References

- [1] Cook, S. A., The Complexity of Theorem Proving Procedures, In *Proceedings of the Third Annual ACM Symposium on the Theory*

*of Computing*, 151-158, 1971.

- [2] Coudert, O. and Madre, J., Implicit and Incremental Computation of Primes and Essential Primes of Boolean Functions, In *Proceedings of the 29th ACM/IEEE Design Automation Conferation*, 36-39, 1992.
- [3] Jackson, P. and Pais, J., Computing Prime Implicants, In *Proceedings of the 10th International Conference on Automated Deduction*, Kaiserslautern, Germany, July. In Lecture Notes in Artificial Intelligence, Springer-Verlag, Vol. 449, 543-557, 1990.
- [4] de Kleer J., An Assumption Based TMS, *Artificial Intelligence*, **28**, 127-162, 1986.
- [5] de Kleer, J., An Improved Incremental Algorithm for Computing Prime Implicates. In *Proceedings of AAAI-92*, San Jose, CA, 780-785, 1992.
- [6] Kean, A., Tsiknis, G., An Incremental Method for Generating Prime Implicants / Implicates, *J. of Symbolic Computation*. **9**, 185-206, 1990.
- [7] Marquis, P., Knowledge Compilation Using Theory Prime Implicates. In *Proceedings of IJCAI*, 837-843, 1995.
- [8] Ngair, T., A New Algorithm for Incremental Prime Implicate Generation. In *Proceedings of IJCAI-93*, Chambery, France, 1993.
- [9] Reiter, R., de Kleer, J., Foundations of Assumption-Based Truth Maintenance System(ATMS): Preliminary Report, In *Proceedings of AAAI-87*, 183-188, 1987.
- [10] Raut, M. K. and Singh, A., Prime Implicates of First Order Formulas, *International Journal of Computer Science and Applications*, **vol. 1(1)** 1-11, 2004.
- [11] Shiny, K., Pujari, A. K., Computations of Prime Implicants using Matrix and Paths, *J. of Logic and Computation*, **8**, 135-145, 1998.
- [12] Slagle, J. R., Chang C. L., Lee, R. C. T., A New algorithm for Generating Prime Implicants, *IEEE trans. on Comp.*, **C-19 (4)**, 304-310, 1970.

- [13] Strzemecki, T., Polynomial-time Algorithm for Generation of Prime Implicants. *Journal of Complexity* **8** , 37-63, 1992.
- [14] Tison, P., Generalized Consensus Theory and Application to the Minimisation of Boolean Functions, *IEEE Trans. on Elec. Comp*, **EC-16** (4), 446-456, 1967.
- [15] del Val, A., Tractable Databases: How to Make Propositional Unit Resolution Complete Through Compilation, In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning*, Bonn, 551-561, 1994.
- [16] del Val, A., Approximate Knowledge Compilation: The First Order Case, AAAI'96, In *Proceedings of AAAI-96*, Portland, Oregon, 498-503, 1996.